WHAT IS CLAIMED IS:

1. A partial merging method for sub-register data operations in a processor, the method comprising:

examining an instruction before execution to identify a portion of a source register identified in the instruction that should remain unchanged into a destination register; and

moving the portion of the source register determined to remain unchanged into the destination register before instruction execution is complete.

2. The method of claim 1, wherein moving the unchanged portion of the source register into the destination register includes setting corresponding source register values to zero.

3. The method of claim 1, wherein the source and destination registers have a greater bit-length than a result bit-length of the instruction.

4. The method of claim 3, wherein the bit-length of the source and destination registers is 32 bits.

5. The method of claim 3, wherein the result bit-length is greater than 1 bit and less than 32 bits.

6. The method of claim 3, wherein the result bit-length is greater than 1 bit and less than 16 bits.

7. A method for sub-register data operations for executing an instruction, the method comprising:

executing the instruction on a first register and a second register; and

merging a result of the executed instruction with a plurality of high-order bits from the first register, the plurality of high-order bits being copied into high-order bit positions of a result register, and the result being placed into low-order bit positions of the result register.

8. The method as recited in claim 7, wherein the merging a result comprises:

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding contents of the first register with the modified second register; and

placing the result in the result register.

9. The method as recited in claim 8, the method further comprising:

ignoring a carryover of the result from the low-order bit positions of the result register to the high-order bit positions of the result register.

10. The method as recited in claim 7, wherein the merging a result comprising:

modifying contents of the first register by placing data values of zero in the low-order bit positions of the first register;

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding the modified first register with the modified second register; and

placing the result in the result register.

11. The method as recited in claim 10, the method further comprising:

ignoring a carryover of the result from the low-order bit positions of the result register to the high-order bit positions of the result register.

12. The method of claim 7, wherein the first register and the second register have 32 bits.

13. The method of claim 7, wherein the result register has 32 bits.

14. The method of claim 7, the method further comprising:

using a renamer to assign the first register, the second register, and the result register.

15. The method of claim 7, wherein the result of the executed instruction is less than 32 bits.

16. The method of claim 15, wherein the result of the executed instruction is less than or equal to 16 bits.

17. The method of claim 16, wherein the result of the executed instruction is less than or equal to 8 bits.

18. The method of claim 7, wherein the merging a result is performed before instruction execution is complete.

19. A processor comprising:

an instruction set having an instruction;

a source register and a destination register referenced by the instruction from the instruction set; and

a logic circuit adapted to examine the instruction before execution to identify a portion of the source register that should remain unchanged into the destination register, and the logic circuit further adapted to move the unchanged portion into the destination register before instruction execution is complete.

20. The processor of claim 19, wherein the logic circuit is adapted to move the unchanged portion into the destination register by setting corresponding values of the source register to zero.

21. The processor of claim 19, wherein the source register and the destination register have a greater bit-length than a result of the instruction.

22. The processor of claim 21, wherein the source register and the destination register have 32 bits.

23. The processor of claim 21, wherein the result of the instruction has less than 32 bits.

24. The processor of claim 23, wherein the result of the instruction is less than or equal to 16 bits.

25. A method for executing instructions in a processor using data registers of different bit lengths while maintaining architecture compatibility, the method comprising:

receiving an instruction to perform an operation on contents of first and second source registers, the contents including a plurality of bits and the operation results being a different bit length then bit lengths of the first and second source registers;

screening the first and second source registers; and

merging the operation results into a destination register when the operation is performed.

26. The method of claim 25, wherein screening the first and second source registers comprising:

identifying high order bits of one of the source registers that should remain unchanged when merged into the destination register; and

modifying the contents of the other source register by setting corresponding high order bits of the other source register to zero.

27. The method of claim 26, wherein merging the operation results comprising:

adding the contents of the one of the source registers with the modified contents of the other source register; and

placing results of the addition in the destination register.

28. The method of claim 27, further comprising:

ignoring a carryover of the addition results from low-order bit positions of the destination register to high-order bit positions of the result register.

29. The method of claim 25, wherein screening the first and second source registers comprising:

modifying contents of one of the source register by setting low order bits of the one of the source registers to zero; and

modifying the contents of the other source register by setting high order bits of the other source register to zero.

30. The method of claim 29, wherein merging the operation results comprising:

adding the modified contents of the one of the source registers with the modified contents of the other source; and

placing results of the addition into the destination register.